



(12) 发明专利

(10) 授权公告号 CN 108920552 B

(45) 授权公告日 2022.04.29

(21) 申请号 201810630231.9

(22) 申请日 2018.06.19

(65) 同一申请的已公布的文献号

申请公布号 CN 108920552 A

(43) 申请公布日 2018.11.30

(73) 专利权人 浙江工业大学

地址 310014 浙江省杭州市下城区朝晖六区潮王路18号

(72) 发明人 杨良怀 卢晨曦 范玉雷

(74) 专利代理机构 杭州浙科专利事务所(普通合伙) 33213

代理人 周红芳

(51) Int.Cl.

G06F 16/22 (2019.01)

(56) 对比文件

CN 102622414 A, 2012.08.01

CN 104899297 A, 2015.09.09

US 2010/0146004 A1, 2010.06.10

CN 106021560 A, 2016.10.12

杨良怀等.一种大数据流内存B+树构建方法.《计算机科学》.2018,第45卷(第3期),第173-179、214页.

Yan Ma等.Task-Tree Based Large-Scale Mosaicking for Massive Remote Sensed Imageries with Dynamic DAG Scheduling.《IEEE》.2013,第2126-2137页.

审查员 赵小娟

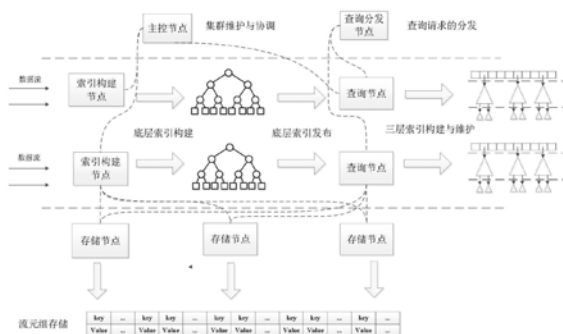
权利要求书2页 说明书10页 附图3页

(54) 发明名称

一种面向多源大数据流的分布式索引方法

(57) 摘要

本发明公开了一种面向多源大数据流的分布式索引方法,可实现数据流高效存储、索引快速构建,并提供高并发的近实时查询服务。针对多源大数据流特点,本发明提出了三层的分布式索引结构:最底层是基于数据流元组的索引,中间层是基于数据流时间窗口的索引,顶层是基于数据源的索引。由于单个计算节点的资源和处理能力有限,无法有效应对潜在规模无限和实时到达的数据流应用场景,本发明设计出主从结构和P2P结构相结合的集群架构。集群中主要由索引构建节点,查询节点,查询分发节点,存储节点,主控节点这五种节点组成,同时考虑集群中的数据存储、负载均衡、可扩展性因素,使得各个节点相互协作,稳定运转,从而提供高效的服务。



1. 一种面向多源大数据流的分布式索引方法,采用三层的分布式索引结构,实现多源大数据流高效存储、快速构建索引,并提供高并发的近实时查询服务,其特征在于具体为采用主从结构和P2P结构相结合的分布式集群架构,并通过将分布式集群架构中不同类型节点的有机组合,实现数据流元组存储、索引构建与维护,并提供高并发的近实时查询服务;所述分布式集群架构包括:

主控节点,负责监控集群整体情况,协调集群中的所有节点;

索引构建节点,负责接收数据流以及索引的构建和发布工作;

存储节点,负责数据流元组的最终存储;

查询节点,负责索引的维护以及响应查询请求;

查询分发节点,负责分发查询请求,控制查询请求的负载均衡;

所述采用三层的分布式索引包括:最底层的基于数据流元组的索引、中间层的基于数据流时间窗口的索引及顶层的基于数据源的索引,索引的构建包括如下步骤:

1) 索引构建节点实时接收数据流元组并将其缓存,对当前时间窗口数据流元组缓存,并构建底层索引;

2) 底层索引构建完成后,向主控节点请求,将底层索引发布到指定的查询节点;

3) 顶层索引的量级小,且发生更新的情况少,对于每一个查询节点,都会维护顶层索引副本;查询节点收到底层索引后,根据接收的数据源标识通过查询节点自身的顶层索引找到对应的中间层索引,并根据时间窗口起始时间向中间层索引插入一条相应的记录,更新中间层索引,依次完成中间层索引和顶层索引的插入构建;

4) 缓存在索引构建节点的数据流元组在特定条件下,由主控节点将数据从构建节点迁移到指定的存储节点上,数据迁移结束后,修改查询节点中底层索引指针。

2. 根据权利要求1所述的面向多源大数据流的分布式索引方法,其特征在于步骤4)中的特定条件指索引构建节点内存紧张、流速较慢的情况具体如下:

I) 当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的a%时,且当前数据流接收流速小于b时,就认为构建节点较为空闲,这时开启数据迁移,把流元组迁移到存储节点;

II) 当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的c%时,就认定构建节点的内存处于紧张的状况,应立即迁移数据,当数据迁移时,数据流的接收速率大于d时,便要通过控制节点将该节点已经接入点的数据源迁移到其他空闲的构建节点,以保证数据迁移的同时不影响数据流的接收和索引的构建;其中,a,b,c,d四个参数根据集群中节点的性能进行统一配置。

3. 根据权利要求1所述的面向多源大数据流的分布式索引方法,其特征在于查询包括以下步骤:

i) 查询分发节点接收到查询请求后,根据负载均衡策略,将查询请求转发到相应的查询节点;

ii) 查询节点接收到查询请求,根据三层索引定位数据流元组位置,若数据流元组不存在,则直接返回查询请求方;若存在,到数据流元组所在存储节点的相应位置获取数据,并返回查询请求方,完成数据查询任务。

4. 根据权利要求3所述的面向多源大数据流的分布式索引方法,其特征在于步骤i)中

的索引构建过程中的负载均衡包括构建节点的负载均衡、存储节点的负载均衡及查询节点的负载均衡；构建节点的负载均衡用于使索引构建节点均衡的分摊数据源，完成索引构建工作；存储节点的负载均衡，用于使数据流元组均匀的存储在存储节点中；查询节点的负载均衡，用于均匀分摊所有的查询请求。

5. 根据权利要求1所述的面向多源大数据流的分布式索引方法，其特征在于所述主控节点维护着所有节点的信息，当集群中的任何一类节点无法维持负载时，能动态增加该类节点，在主控节点中注册新增节点，增加节点后，主控节点会启动负载均衡策略，动态调整节点的负载，从而使得集群稳定运转。

一种面向多源大数据流的分布式索引方法

技术领域

[0001] 本发明涉及多源大数据流分布式存储、索引与查询技术，尤其涉及一种面向多源大数据流的分布式索引方法。

背景技术

[0002] 近年来，数据流的应用越来越广泛。典型的数据流包括传感网感知数据、工业实时采集数据流、股价数据、网络监控数据、交通监控系统的监测数据、电信通话记录以及Web点击流。数据流是由流元组构成的一个无限的有序序列。本发明中流元组的形式为 $\langle ts, tuple \rangle$ ，其中“ts”指流元组的时间戳，“tuple”指数据元组。其中，流元组往往来自多个源头，“tuple”中也会记录流元组源头的相关信息。与传统的数据管理不同，数据流具有以下的特点：(1) 数据流连续、实时到达；(2) 数据流的潜在规模是无限的；(3) 系统无法控制数据流到达的速率和元组的顺序；(4) 数据流一经处理，除非特意保存，否则难以再次处理。传统数据库管理系统在处理规模有限、数据更新频率相对低的数据集合时具有良好的性能；但面对数据流更新持续且频繁，在很短时间内（秒级甚至毫秒级）要处理大量的数据更新，传统数据库无法维持这样的更新频率。

[0003] 流数据处理方面已有大量的研究工作。研究人员设计了一些适用于数据流处理的系统，如斯坦福大学的STREAM系统，加州大学伯克利分校的TelegraphCQ系统，麻省理工学院、布朗大学和布兰代斯大学联合研发的Aurora系统。上述流数据库系统目的是用于支持在线数据的静态查询、连续查询、近似计算等，不在磁盘上保存整个数据流，仅保存查询结果（如数据聚集）。这些系统通常提供了支持流数据操作的类SQL语言。近几年，数据流应用不再局限于流数据实时处理的速度和查询结果的精确度方面的高要求，这些应用也拓广到海量流数据的深度分析上，对数据流的实时存储提出了要求。而上述数据流管理系统没有关注数据流的实时存储问题。

[0004] 数据实时性、突发性、无限性的特点对数据流的存储速度提出了很高的要求。数据流的实时存储涉及到数据流上实时构建合适的索引，方便后续即时查询操作。有研究人员对网络数据包流的特殊应用提出了一种实时位图索引，针对传感器网络中异构数据流构建位图索引。上述工作针对特定的数据流应用提出的索引构建方法，还无法涵盖各种场景。

[0005] 针对时间窗口的数据流是重要的应用场景之一，如交通数据流，需要对每个时间段的数据进行处理、分析、存储，本发明可以处理该场景下的数据流。由于数据流实时性、无限性的特点，单台机器无法完成索引构建、数据存储、响应实时查询请求这些需求，所以本发明提出分布式场景下的解决方案。

发明内容

[0006] 针对现有技术中存在的以上问题，本发明提出一种面向多源大数据流的分布式索引方法，实现多源大数据流高效存储，对海量数据流快速构建索引，并提供高并发的近实时查询服务。

[0007] 所述的一种面向多源大数据流的分布式索引方法,采用三层的分布式索引结构,实现多源大数据流高效存储、快速构建索引,并提供高并发的近实时查询服务,其特征具体为采用主从结构和P2P 结构相结合的分布式集群架构,并通过将分布式集群架构中不同类型节点的有机组合,实现数据流元组存储、索引构建与维护,并提供高并发的近实时查询服务;所述分布式集群架构包括:

[0008] 主控节点,负责监控集群整体情况,协调集群中的所有节点;

[0009] 索引构建节点,负责接收数据流以及索引的构建和发布工作;

[0010] 存储节点,负责数据流元组的最终存储;

[0011] 查询节点,负责索引的维护以及响应查询请求;

[0012] 查询分发节点,负责分发查询请求,控制查询请求的负载均衡。

[0013] 所述的面向多源大数据流的分布式索引方法,其特征采用三层的分布式索引包括:最底层的基于数据流元组的索引、中间层的基于数据流时间窗口的索引及顶层的基于数据源的索引。

[0014] 所述的面向多源大数据流的分布式索引方法,其特征索引的构建包括如下步骤:

[0015] 1) 索引构建节点实时接收数据流元组并将其缓存,对当前时间窗口数据流元组缓存,并构建底层索引;

[0016] 2) 底层索引构建完成后,向主控节点请求,将底层索引发布到指定的查询节点;

[0017] 3) 顶层索引的量级小,且发生更新的情况少,对于每一个查询节点,都会维护顶层索引副本;查询节点收到底层索引后,根据接收的数据源标识通过查询节点自身的顶层索引找到对应的中间层索引,并根据时间窗口起始时间向中间层索引插入一条相应的记录,更新中间层索引,依次完成中间层索引和顶层索引的插入构建;

[0018] 4) 缓存在索引构建节点的数据流元组在特定条件下,由主控节点将数据从构建节点迁移到指定的存储节点上,数据迁移结束后,修改查询节点中底层索引指针。

[0019] 所述的面向多源大数据流的分布式索引方法,其特征步骤4) 中的特定条件指索引构建节点内存紧张、流速较慢的情况具体如下:

[0020] I) 当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的a%时,且当前数据流接收流速小于b时,就认为构建节点较为空闲,这时开启数据迁移,把流元组迁移到存储节点;

[0021] II) 当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的c%时,就认定构建节点的内存处于紧张的状况,应立即迁移数据,其中,a,b,c,d四个参数根据集群中节点的性能进行统一配置。

[0022] 所述的面向多源大数据流的分布式索引方法,查询包括以下步骤:

[0023] i) 查询分发节点接收到查询请求后,根据负载均衡策略,将查询请求转发到相应的查询节点;

[0024] ii) 查询节点接收到查询请求,根据三层索引定位数据流元组位置,若数据流元组不存在,则直接返回查询请求方;若存在,到数据流元组所在存储节点的相应位置获取数据,并返回查询请求方,完成数据查询任务。

[0025] 所述的面向多源大数据流的分布式索引方法,其特征步骤i) 中的索引构建

过程中的负载均衡包括构建节点的负载均衡、存储节点的负载均衡及查询节点的负载均衡；构建节点的负载均衡用于使索引构建节点均衡的分摊数据源，完成索引构建工作；存储节点的负载均衡，用于使数据流元组均匀的存储在存储节点中；查询节点的负载均衡，用于均匀分摊所有的查询请求。

[0026] 所述的面向多源大数据流的分布式索引方法，其特征在于所述主控节点维护着所有节点的信息，当集群中的任何一类节点无法维持负载时，能动态增加该类节点，在可控节点中注册新增节点，增加节点后，主控节点会启动负载均衡策略，动态调整节点的负载，从而使得集群稳定运转。

[0027] 通过采用上述技术，与现有技术相比，本发明的有益效果如下：

[0028] 1) 本发明通过采用上述技术，针对多源大数据流特点，提出了三层的分布式索引结构，且采用主从结构和P2P结构相结合的集群架构；集群中主要由索引构建节点，查询节点，查询分发节点，存储节点，主控节点这五种节点组成，同时考虑集群中的数据存储、负载均衡、可扩展性因素，使得各个节点相互协作，稳定运转，可实现数据流高效存储、索引快速构建，并提供高效的近实时查询服务，集群中的节点互相协调但没有完全耦合，所有类型的节点都可以进行横向扩展，使得集群有良好的扩展性；

[0029] 2) 本发明针对于索引构建节点，当数据流流速过大时，索引构建的速度跟不上数据流到达的速度时，集群就不能正常运转，此时，可以动态的添加索引构建节点来降低每个节点的负载，添加过程只需向控制节点注册即可，针对于存储节点，可以动态增加数据存储节点，以应对海量数据存储场景，同样，添加存储节点只需向控制节点注册。对于查询节点，可以动态向查询分发节点注册查询节点来增加查询节点数量，以应对高并发海量查询请求，提高其灵活性；

[0030] 3) 本发明在在集群中，主控节点维护着所有节点的信息，当集群中的任何一类节点无法维持负载时，就可以动态增加该类节点，在控制节点中注册新增节点即可。增加节点后，控制节点会启动负载均衡的策略，从而使得集群稳定运转。

附图说明

[0031] 图1是本发明的分布式索引系统总架构示意图；

[0032] 图2是本发明的三层分布式索引结构示意图；

[0033] 图3是本发明的三层索引结构中的底层索引结构示意图；

[0034] 图4是本发明的三层索引结构中的中间层索引结构示意图；

[0035] 图5是本发明的三层索引结构中的顶层索引结构示意图；

[0036] 图6是本发明的分布式索引方案架构及数据流向示意图。

具体实施方式

[0037] 以下结合说明书附图及实施例对本发明作进一步的描述：

[0038] 如图1所示，本发明的面向多源大数据流的分布式索引方法，采用三层的分布式索引结构，实现多源大数据流高效存储、快速构建索引，并提供高并发的近实时查询服务，其特征在于具体为采用主从结构（主从结构为一个为主节点，其他为从属节点，一种分布式计算的结构）和P2P结构（P2P结构为对等网结构，每个节点都平等）相结合的分布式集群架构，

并通过将分布式集群架构中不同类型节点的有机组合,实现数据流元组存储、索引构建与维护,并提供高并发的近实时查询服务;所述分布式集群架构包括:主控节点、存储节点、索引构建节点、查询节点及查询分发节点,所述主控节点,负责监控集群整体情况,协调集群中的所有节点,维护着所有节点的信息,当集群中的任何一类节点无法维持负载时,能动态增加该类节点,在所述主控节点中注册新增节点,增加节点后,所述主控节点会启动负载均衡策略,动态调整节点的负载,从而使得集群稳定运转;

[0039] 所述存储节点,负责数据流元组的最终存储;

[0040] 所述索引构建节点,负责接收数据流以及底层索引的构建和发布工作;

[0041] 所述查询节点,负责三层索引的维护与查询请求的响应,包括中间层索引和顶层索引的构建;

[0042] 所述查询分发节点,负责分发查询请求,控制查询请求的负载均衡。

[0043] 如图2所示,本发明的分布式索引方案采用了三层分层构建的分布式索引结构,其最底层是基于数据流元组的索引,中间层是基于数据流时间窗口的索引,顶层是基于数据源的索引,下面对每层索引进行详细说明:

[0044] 如图2所示,本发明的分布式索引方案采用了三层分层构建的分布式索引结构,其最底层是基于数据流元组的索引,中间层是基于数据流时间窗口的索引,顶层是基于数据源的索引,下面对每层索引进行详细说明:

[0045] 1) 底层索引基于每个数据源的每个时间窗口中流元组内容的索引,选用B+树索引结构。B+树同时支持单点和范围查询,在大数据量的情况下,B+树层高小的特点能够减少数据访问的层数,减小查询处理的开销。一个数据源的一个时间窗口内的数据流元组构建成一棵底层B+树,如图3所示。底层B+树的索引项<key, ptr>的“key”值是数据流元组中的“key”值,内部节点的“ptr”指向是B+树子节点,叶节点的“ptr”指向流元组存储在内存中的位置。其中,在分布式的系统中,流元组和底层索引不在同一个节点,故叶子结点的“ptr”可以用流元组所在结点的ip+内存地址表示,然而这种方式会导致“ptr”字段的长度过长,浪费内存空间,而且对“ptr”的更新也不方便。由于一个时间窗口的流元组一定放在同一个节点,在内存中是连续存放且对于“key”有序,所以对于一棵底层B+树,使用ip字段记录流元组所在结点的ip,使用起始地址字段记录连续存放的流元组起始内存地址,这样“ptr”只要记录对应流元组对于起始内存地址的偏移量即可,节省内存开销,也便于更新;

[0046] 每个数据源的数据流都是源源不断,快速的到达,构建底层B+树索引方法需要针对数据流特点,提高吞吐、降低时延,快速完成。传统的B+树构建方法中,由于节点要不断的分裂,移动,构建效率低,无法适应数据流的特性;

[0047] 本发明采用基于多微批量排序单批量装载的B+树构建方法快速构建B+树。该方法对一个时间窗口进一步分片处理,分别对每一个分片的流元组进行排序,最后采取归并的方法完成流元组排序。然后根据时间窗口中流元组的数量,计算B+树结构参数,根据计算结果统一申请内存资源,然后自底向上自左向右并行批量装载节点,最后再自底向上自左向右并行操作节点,从其子节点或所属流元组获取码值,从而完成底层索引的构建。底层索引在构建过程中对“key”值进行了排序,所以能快速记录底层索引中“key”值的最大和最小值,作为底层索引的统计信息,用于中间层索引的构建;

[0048] 2) 中间层索引基于每个数据源时间窗口的索引。对于一个数据源,底层构建的索

引局限于一个时间窗口的数据流元组,中间层索引则是针对数据源中所有的时间窗口,中间层索引同样需要支持单点和范围查询,所以选取的也是B+树的索引结构。中间层索引将时间窗口时间作为“key”值,内部节点索引项<key,ptr>中的“ptr”指向中间层索引子节点,叶节点的“ptr”指向与“key”值对应的某个时间窗口的底层B+树根节点,其中中间层索引和底层索引可能不在同一个节点,所以“ptr”由对应底层索引所在结点ip地址+底层索引根节点内存地址组成。中间层索引结构如附图4所示。其中,中间层B+树的叶子节点中,包含对应底层B+树的统计信息,如底层索引“key”的最小值和最大值,便于在查询时过滤掉一些不必要的查询;

[0049] 中间层索引不同于底层索引,对树的更新速率远比底层低,因此采取普通的B+树构建方法;

[0050] 3) 顶层索引基于某种应用语义映射形成索引,如某一地理区域的交通卡口数据。在不同的应用场景中,顶层索引的“key”值和索引结构可以自定义选择,来满足不同场景的不同需求。对于每个数据源,底层和中间层索引组成了一颗双层的B+树索引。顶层索引是针对所有数据源,将中间层B+树根节点地址作为一个“value”值,以数据源标识信息作为“key”值,形成<key,value>元组。其中,顶层索引和中间层索引可能不在同一个节点,故“value”值由对应中间层层索引所在结点ip地址+中间层索引根节点内存地址组成;

[0051] 由于中间层索引随着时间的推移,B+树不断的增大,B+树的根节点也会发生变化。顶层索引的“value”也随之改变,不利于索引的维护。为了固定顶层索引,在中间层索引中增加一个固定节点,存放指向中间层B+树的根节点的指针,顶层索引的value指向该固定节点。当中间层B+树根节点变化时,只需调整固定节点的指针,无需改变顶层索引。

[0052] 如图1所示,本发明的分布式集群中包括主控节点、索引构建节点、存储节点、查询节点、查询分发节点。整个集群是主从结构,主控节点相当于集群中的主节点,监控其他的所有节点,协调集群中的所有节点,其他节点相当于从属节点,通过和主控节点的协商完成各自的任务。下面对每一类节点进行详细说明:

[0053] 索引构建节点负责数据流接收,底层索引构建与发布,主要有以下功能。

[0054] (1) 实时接收数据流数据,缓存时间窗口的数据流元组。针对时间窗口的元组数据快速构建底层B+树索引;

[0055] (2) 将构建完成的B+树索引、B+树对应的时间窗口起始时间以及数据源标识发送给查询节点。考虑负载均衡,由主控节点选择一个查询节点,并把索引发送到该查询节点中去。由于索引中存在很多的指针,所以在发送的过程中,索引节点在内存中是连续的,把索引内部的指针转换为对于根节点的偏移量,这样查询节点在收到偏移量后,能快速的重构底层索引;

[0056] (3) 为了提高索引构建效率,流元组数据一开始是暂存在构建节点的,随着时间的推移,流元组的数据量巨大,构建节点在特定的条件下要将流元组数据转移到存储节点中;

[0057] (4) 数据流元组缓存在构建节点的这段时期,构建节点会响应对该部分数据的查询请求;

[0058] 查询节点主要负责三层索引的维护与查询的处理,主要有以下功能:

[0059] (I) 接收索引构建节点发送的底层B+索引结构,把索引节点的偏移量重新转化为具体的指针,完成B+树的重建。根据接受的数据源标识通过查询节点自身的顶层索引找

到对应中间层索引的所在节点,并根据时间窗口起始时间向中间层索引插入一条记录。

[0060] (II) 查询节点维护整个的三层索引结构。其中,由于顶层索引的量级小,且发生更新的情况较少,所以对于每一个查询节点,都会维护顶层索引,这样保证查询请求不管落在哪个节点上,都可以向下查询,从而提高查询效率。

[0061] (III) 查询节点实时的接收查询分发节点分发的查询请求,根据三层的索引结构,定位到数据所在的节点位置,并向该节点请求获取数据。

[0062] 存储节点负责数据流元组的最终存储,主要有以下功能。

[0063] (i) 接受从索引构建节点迁移过来的数据流元组,数据流元组最终存储在存储节点上,数据存储节点可以根据数据流规模的持续增长,动态扩展节点数量,以提供海量数据存储服务。数据迁移的过程中,将索引构建节点缓存的数据流拷贝一份迁移到存储节点,存储节点会更新底层索引对应的ip字段和流元组起始位置字段;

[0064] (ii) 响应流元组数据的查询请求

[0065] 主控节点负责监控集群整体情况,协调集群中的所有节点。主要有以下功能:

[0066] (a) 主控节点中维护着在线的其他所有节点信息,节点信息包括内存利用率、cpu利用率等信息。从而可以监控各种节点;

[0067] (b) 集群中的所有节点与主控节点会定时保持心跳,并更新对应的节点信息,节点的上下线可以被主控节点所感知;

[0068] (c) 数据迁移时,主控节点会根据所有存储节点的信息,选择负载较小的节点进行迁移,从而达到负载的均衡。索引发布时,主控节点选取负载较小的查询节点进行索引发布。

[0069] 查询分发节点负责将查询请求均衡的分发到查询节点,从而提高系统的查询效率。

[0070] 本发明提出的分布式内存索引通过这五种不同类型节点的有机组合,实现数据流元组存储、三层索引构建与维护,并提供高并发的近实时查询服务。

[0071] 如图所示,本发明的分布式索引构建方法,其步骤主要有:

[0072] 1) 索引构建节点实时接收数据流元组,缓存当前时间窗口数据流元组,用上文提到的批量构建B+树的方法快速的构建底层索引;

[0073] 2) 底层索引构建完成后,要将底层索引发布到查询节点。同时,将该B+树对应时间窗口的起始时间戳以及数据流元组对应的数据源标识发送给查询节点,用于上层索引的构建。发布索引要考虑查询节点的负载均衡,由主控节点根据负载均衡策略选择一个查询节点进行发布。由于底层索引结构中存在很多的指针,直接发布到查询节点会造成索引的不可用。所以在发布的过程中,要先将指针进行转化。由于索引节点在内存中是连续存放的,索引可以把索引内部的指针转换为基于根节点位置的偏移量,这样查询节点在收到偏移量后,根据实际根节点地址,把偏移量转换成实际的指针,从而快速的重建底层索引;

[0074] 3) 查询节点收到构建节点发送的底层B+树后,根据接受到的节点数组以及各节点的对应于根节点的偏移,与根节点的实际地址相加得到指针应该指向的实际地址,这样就可以快速重建底层索引。此时,数据流元组和底层索引不在同一个节点上,所以要将底层索引中的ip 字段值改为流元组在节点的ip;

[0075] 4) 查询节点根据收到的数据源标识以及查询节点自身的顶层索引找到该数据源

对应的中间层索引,并把时间窗口起始时间作为“key”,对应的底层索引的地址(ip+内存地址)作为“value”插入到中间层索引中,若中间层索引的根节点发生了变化,则改变中间层索引的固定节点的指针,使它指向新的根节点位置。若顶层索引中的“key”值中没有对应的数据源标识,则表明该段数据是该数据源的第一个时间窗口的数据,查询节点会向主控节点申请添加数据源,主控节点会选择一个查询节点用于构建该数据源对应的中间层索引,把底层索引时间窗口起始时间作为“key”,根节点地址(ip+内存地址)作为“value”插入到新建的中间层索引,把中间层索引中固定节点的指针指向当前的根节点地址,并把选定查询节点的ip和固定节点的内存地址做为“value”插入到顶层索引。由于各个查询都维护着一份顶层索引,则主控节点要将顶层索引的更新操作同步到其他所有的查询节点。为了保证顶层索引的强一致性,当顶层索引同步成功后,才算整个构建过程完成;

[0076] 5) 索引构建节点在接收数据流元组后,会缓存数据流元组,一定条件下,构建节点会把数据流元组迁移到存储节点。索引构建节点在达到数据迁移条件后,就会启动数据迁移流程。首先,向控制节点请求数据迁移,控制节点根据存储节点的负载情况选取一个存储节点进行迁移。其中,系统中存储的最小单元是一个数据源的一个时间窗口的数据,这也作为数据迁移时的最小单元;

[0077] 6) 索引节点将暂存的数据发送到指定的存储节点中,存储节点接收后,由于流元组的位置发生了改变,故需要修改对应底层索引叶子节点的指针。存储节点通过主控节点随机选取一个查询节点,通过数据源标识和起始时间窗口根据三层索引结果找到对应数据源时间窗口的底层索引,将底层索引对应的ip字段改为实际存储节点的ip地址,将存储起始位置字段改为流元组实际存储的起始内存位置。修改完成后,删除构建节点缓存的流元组。从而完成数据迁移。

[0078] 通过以上六步,完成分布式索引的构建。

[0079] 如图所示,本发明的索引查询流程,其步骤主要有:

[0080] i) 查询分发节点接受到请求后,根据负载均衡策略,将请求转发到相应的查询节点;

[0081] ii) 查询节点接收到查询请求,根据顶层索引查找到指数数据源对应的中间层索引,并将查询请求转发到中间层索引所在的查询节点中。根据中间层索引,找到对应时间窗口的叶子结点,此时可以结合索引叶子节点的统计信息,如果要查询的值不在最小值和最大值的区间内,就不再继续查找底层索引,并返回查询失败消息。反之,则获取底层索引存储位置信息,把请求转发到底层索引对应的节点。再查询底层 B+树索引,找到对应的基于流元组起始位置的偏移量,根据底层索引的ip值以及对应的流元组的起始位置,定位所要查找元组的存储位置,转发请求以获取流元组数据;

[0082] iii) 数据所在的节点(如果数据没有转移,数据在索引构建节点。如果数据已经转移,数据在存储节点)根据请求找到数据,并返回给查询请求方,完成数据查询任务;

[0083] 通过以上三步,完成系统提供的查询服务。

[0084] 本发明在索引构建过程中采用的数据存储策略为了适用于数据规模无限,且有近实时查询需求的数据流应用场景,本方案所采用的存储策略是将分布式索引和数据流元组存储在可动态扩展的集群中,以解决海量数据高效存储问题。核心思想是将数据流元组与分布式索引分离,存储在不同的节点类型中。为了提高索引的构建效率,数据流元组一开始

缓存在索引构建节点中,在特定的条件下触发数据迁移,将流元组从构建节点迁移到存储节点中。在以下的两种情况下,会触发数据迁移:

[0085] I) 索引构建节点在构建完索引后,数据流元组暂存在构建节点,当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的 $a\%$ 时,且当前数据流接收流速小于 b 时,就认为构建节点较为空闲,这时开启数据迁移,把流元组迁移到存储节点;

[0086] II) 当索引构建节点缓存的数据流元组所占的内存达到索引节点内存的 $c\%$ 时,就认定构建节点的内存处于紧张的状况,应立即迁移数据。当数据迁移时,数据流的接收速率大于 d 时,便要通过控制节点将该节点已经接入点的数据源迁移到其他空闲的构建节点,以保证数据迁移的同时不影响数据流的接收和索引的构建;

[0087] 其中, a, b, c, d 四个参数需要根据集群中节点的性能情况做统一配置,具体如下:

[0088] 参数 a 和 c 根据索引构建节点的内存大小和数据流的当前预计平均流速决定,参数 a 需要保证剩余的内存空间还能支撑大约5个时间窗口左右的数据流,参数 c 需要保证剩余的内存空间还能支撑大约2个时间窗口左右的数据流。由于数据流速存在着一定的波动,每隔几个时间窗口周期性的更新平均流速,并更新相应的参数 a 和 c 。

[0089] 参数 b 和 d 主要根据机器的性能进行决定。在流速 b 的情况下,构建节点构建一个底层索引的时间为时间窗口的时长一半左右;在流速 d 的情况下,构建节点构建一个底层索引的时间为时间窗口的时长五分之四左右。事先根据节点的性能情况,通过测试得到参数 b 和 d ;

[0090] a, b, c, d 四个参数通过配置文件的形式,在集群中的每个构建节点上保存一份,构建节点根据配置文件进行相应的逻辑处理,其中 a 和 c 两个参数需要定时的进行更新。

[0091] 本发明在构建过程中负载均衡策略,负载均衡是集群需要考虑的重要环节,实现负载均衡能够使得集群更加稳定,资源的利用率也更高。本发明设计的集群中,需要考虑三个方面的负载均衡,分别是构建节点的负载均衡,存储节点的负载均衡,查询节点的负载均衡,下面对这三个方面做详细的介绍:

[0092] 构建节点主要负责数据流的接收和索引的构建,通常情况下,一个构建节点会接收多个数据源的数据流。如果某些构建节点的所连接的数据源的数据流流速过大,将严重影响索引构建的效率,甚至导致构建节点崩溃而不可用。所以要尽量保证每个构建节点所接收的数据流流速要大致均衡。构建节点要和控制节点保持心跳,每隔时间 T_1 向控制节点发送节点的当前的数据流流速。控制节点根据当前节点的流速,进行动态调整。若某个构建节点的数据流流速大于 S_1 ,就将该构建节点中接入的数据源转移到当前数据流流速最小的构建节点上,接入数据源的转移要等到数据源的当前时间窗口的数据接收完毕后再进行转移,以保证每个数据源的每个时间窗口的数据在同一个构建节点进行索引构建。其中, T_1 和 S_1 两个参数要根据集群中节点的性能做统一配置。

[0093] 存储节点主要负责数据流元组最终的存储,存储的最小单元是每个数据源的每个滑动窗口的数据。如果某个存储节点存储的数据流元组数据量过大,查询落在该节点上的比例变高,导致集群中的存储节点失衡,不利于集群的稳定。所以存储节点也要和控制节点保持心跳,每隔时间 T_2 向控制节点发送当前节点的存储情况,在构建节点数据迁移时,控制节点根据集群中存储节点的存储情况,选择把数据迁移到负载较小的存储节点中。其中,参数 T_2 要根据集群中节点的性能做统一配置。

[0094] 如图3-5所示,查询节点负责构建三层索引结构,其中顶层索引和中间层索引的数据量较小且分布情况相对固定,底层索引的数据流与数据源的数据流速有着直接的关系。所以需要考虑底层索引的负载均衡。查询节点和控制节点保持心跳,每隔时间 T_3 向控制节点发送当前节点的底层索引的存储情况,在索引发布的时候,控制节点根据查询节点存储情况,选择把底层索引发布到存储负载较小的查询节点中。其中,参数 T_3 要根据集群中节点的性能做统一配置。

[0095] 查询节点还有一个作用是响应查询请求,如果某个节点的查询请求数量过大,查询的响应速度会变慢,更有可能使得该节点不可用。查询节点的负载均衡通过查询分发节点进行控制,查询分发节点可以通过轮询的方式,将查询请求均衡的分发到查询节点,如果查询节点的性能有较大的差别,分发节点在分发时可以有权重的进行轮询,从而达到负载的均衡。

[0096] 本发明分布式索引方法中的扩展性策略扩展性是指在分布式系统中,数据量,请求数量急剧增加,当前集群的节点数量无法维持负载时,能动态增加节点数量以维持系统稳定运行的能力。对于应用于的大数据流应用场景集群,可扩展性是至关重要的。

[0097] 实施例:下面以城市区域卡口交通数据为例,对本发明的多源大数据流分布式索引方法,从索引的构建,数据的存储,查询的响应这三个方面作进一步说明。

[0098] 卡口交通数据流符合 $\langle ts, tuple \rangle$ 的流元组形式,其中“ts”表示车辆信息记录的时间戳,“tuple”主要包括三种内容:车牌信息、车辆具体信息、监控卡口信息;对于卡口交通数据,可以用卡口的编号作为“key”,选取hash索引作为顶层索引,如附图5所示。

[0099] 下面结合附图6,说明索引构建过程:

[0100] 第一步,选取合适的时间窗口构建底层索引。可以综合卡口的车流量等信息选取合适的时间窗口大小,这里以5分钟为例;如图中①标识的带箭头实线,索引构建节点会缓存5分钟的所有车辆数据流元组,并根据卡口编号把车辆数据流分开,构建节点把车辆车牌信息做为key值,针对5分钟内缓存的数据构建底层B+树索引。假设构建的底层B+树索引对应的时间窗口起始时间为1512455426,对应的卡口编号为k1;

[0101] 第二步,如图中②标识的带箭头实线,构建完成后,构建节点向主控节点申请发布索引,主控选择一个查询节点,构建节点将底层索引发布到该查询节点,查询节点接收底层索引后,根据卡口编号k1,查询顶层索引,若顶层索引存在key为k1的项,则根据value 找到中间层索引,并把key为1512455426,value为底层索引根节点地址的项插入中间层索引;若顶层索引不存在key为k1的项,则新建一棵中间层索引,把key为1512455426,value为底层索引根节点地址的项插入新建的中间层索引,并对顶层索引添加key为k1的项,最后在所有查询节点同步顶层索引更新操作;

[0102] 第三步,卡口编号为k1,时间窗口起始时间为1512455426的车辆数据流元组一开始全部缓存在索引构建节点,当触发数据迁移条件时,索引构建节点会向主控节点请求数据迁移,如图中④⑤⑥标识的带箭头粗实线,主控节点根据负载均衡策略,选取合适的存储节点,并把数据流元组迁移到该存储节点中,迁移完毕后,更改k1,时间窗口起始时间为1512455426对应的底层索引,并删除构建节点缓存的数据流元组;

[0103] 至此,即是卡口交通数据流一个时间窗口数据的完整索引构建过程。

[0104] 假定用户的查询的请求是:查询牌照为a1的车辆在1512455623 时刻前后5分钟内

是否在卡口k1中出现?若出现,则返回车辆的具体信息,否则返回“未查询到此车辆”,下面结合附图6,具体说明索引查询过程:

[0105] 第一步,如图中⑦标识的带箭头虚线,查询分发节点收到查询请求后,根据查询节点的负载情况,考虑负载均衡策略,选取一个查询节点,将请求转发到该查询节点;

[0106] 第二步,如图中⑧标识的带箭头虚线,查询节点使用k1这个key 值,根据顶层hash索引进程查询,若没查找到,直接返回未查询到此车辆,若查找到,根据指针将查询转发到指定查询节点,根据时间戳1512455623、时间窗口时长以及前后5分钟这些条件,查找1512455623所在时间窗口以及它的前一个窗口和后一个窗口,根据中间层索引进行查找;若所查时间窗口均不在所有时间窗口范围内,直接返回未查询到此车辆,若在范围内,依次找到所对应的时间窗口,根据value值的指针,将查询转发到指定查询节点;

[0107] 第三步,如图中⑨标识的带箭头虚线。根据对应的底层索引进行查找,若没有key为a1的节点,直接返回未查询到此车辆,若查找到,根据对应的指针将请求发到对应的存储节点或索引构建节点,获取对应的具体车辆信息并返回给查询用户。

[0108] 至此,即是一个用户请求的完整查询响应过程。

[0109] 上述各实施例仅用于说明本发明,其中各步骤都是可以有所变化的,凡是在本发明技术方案的基础上进行的等同变换和改进,均不应排除在本发明的保护范围之外。

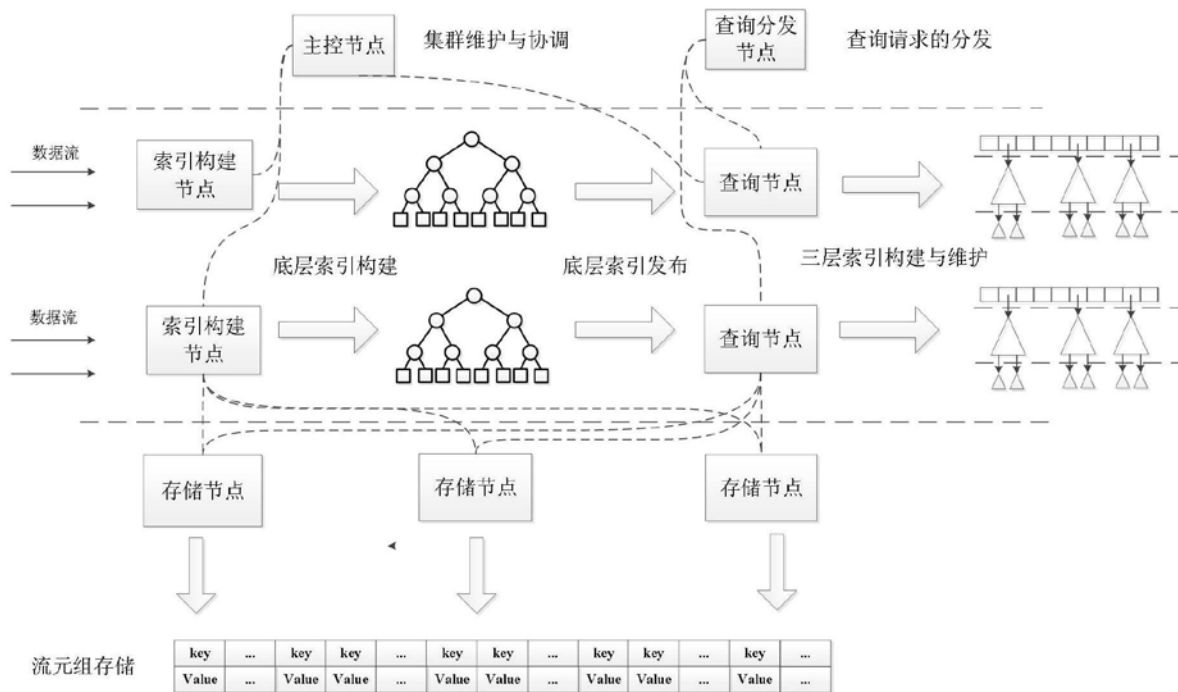


图1

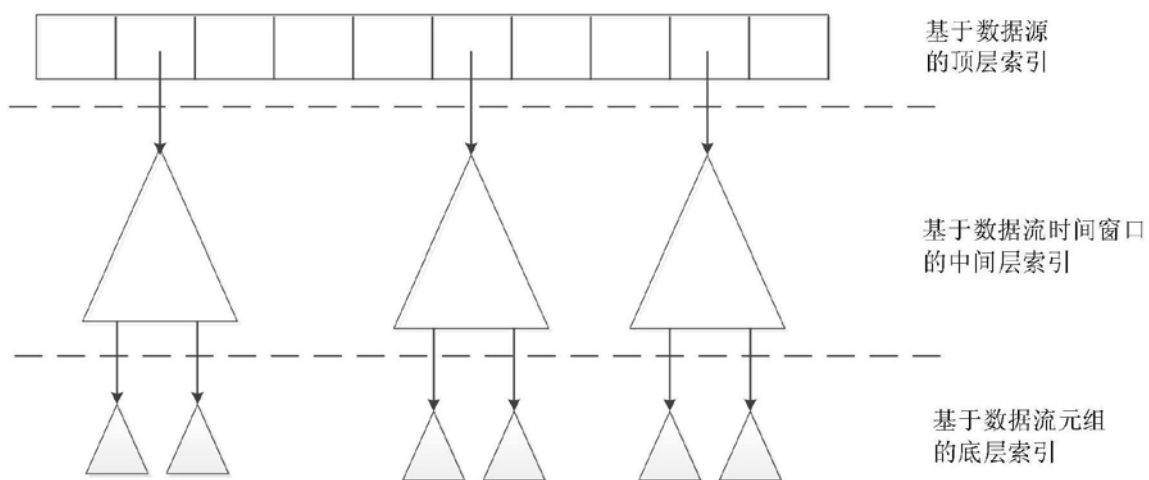


图2

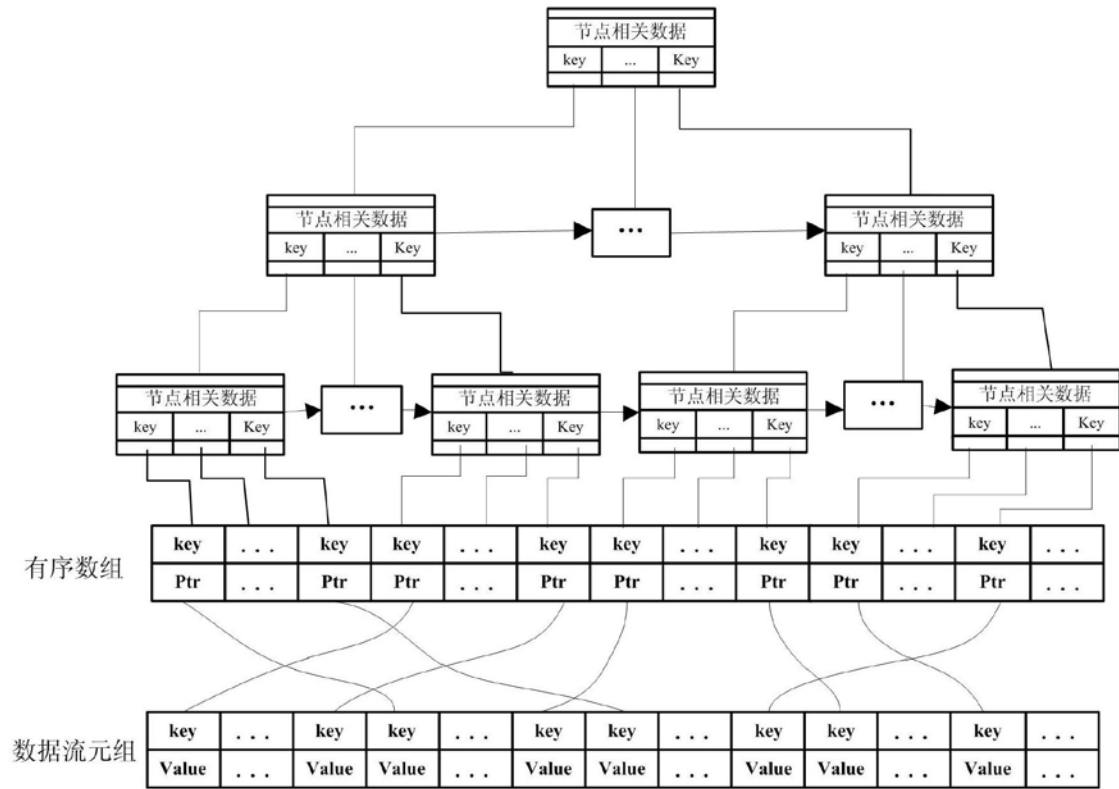


图3

中间层B+树索引

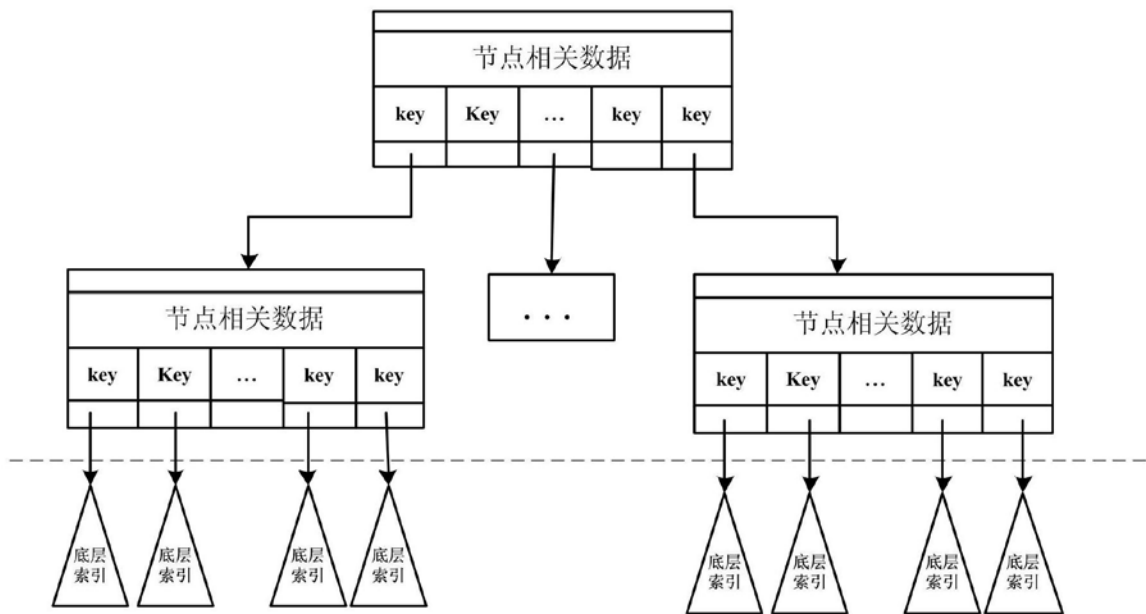


图4

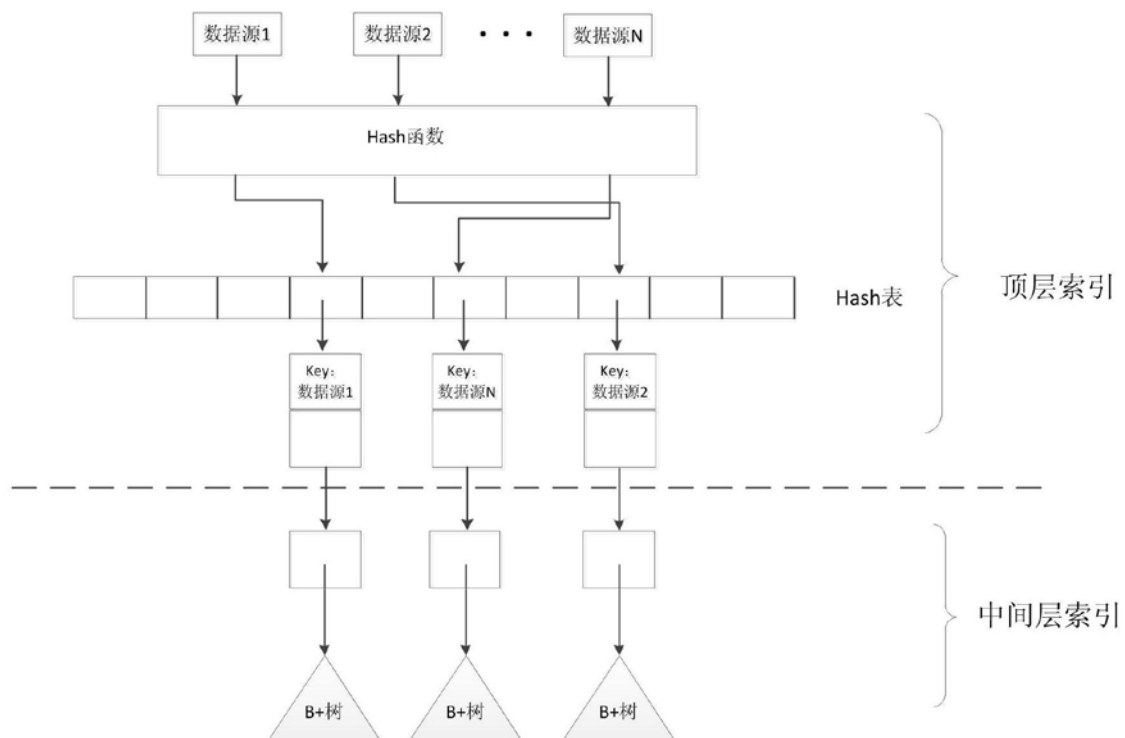


图5

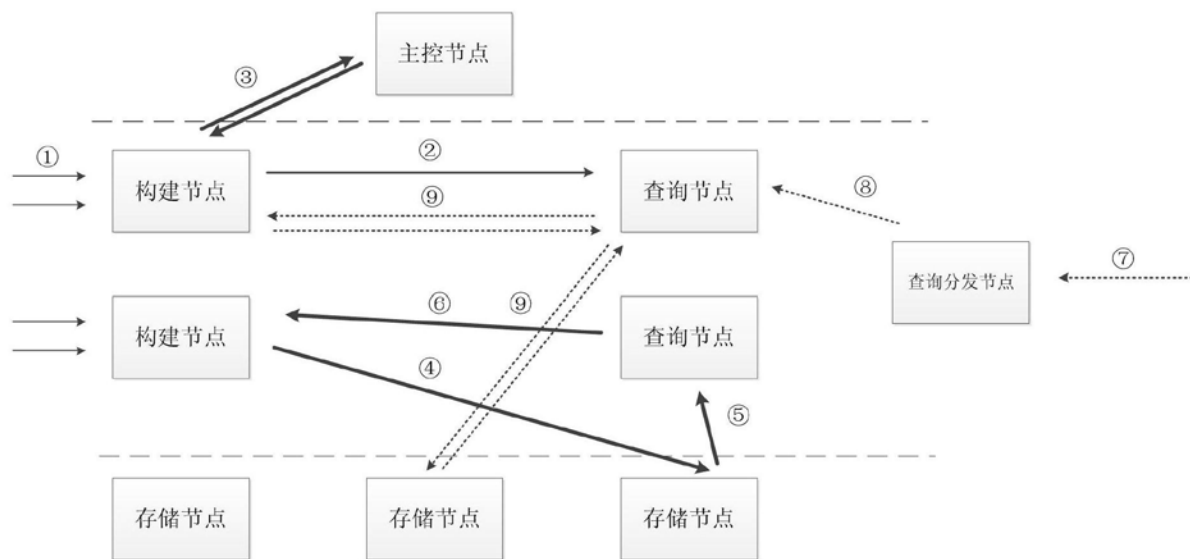


图6